OPT2024: 16th Annual Workshop on Optimization for Machine Learning

Consensus Based Optimization Accelerates Gradient Descent

Anagha Satish ASATISH@CALTECH.EDU
Ricardo Baptista RSB@CALTECH.EDU
Franca Hoffmann FRANCA.HOFFMANN@CALTECH.EDU
California Institute of Technology

Abstract

We propose a novel algorithm for integrating gradient information into Consensus Based Optimization (CBO), a recently proposed multi-particle gradient-free optimization method. During each iteration, a subset of particles are updated using local gradient information, while others are updated using a traditional CBO step. We propose a method for subset selection and investigate its empirical performance. The algorithm combines gradient and gradient-free optimization to encourage exploring the state space while maintaining fast convergence. We investigate the tradeoff between accuracy and computational cost when adjusting the number of gradient evaluations. When applied to classification tasks in machine learning, the proposed algorithm attains a similar accuracy to ensemble gradient methods based on Gradient Descent or Adam at a reduced computational cost.

1. Introduction

Gradient-based optimization algorithms have been at the center of advancements in optimization. For example, stochastic Gradient Descent (SGD)[36] and Adam [28] have become core parts of the machine learning optimization toolbox. These algorithms allow for convergence to an optimal solution by limiting the search to promising directions, while encouraging efficient compute. Advancements in deep learning however, have called for the effective optimization of highly nonconvex functions [38]. Training a neural network involves minimizing a non-convex loss function $f\colon \mathcal{D} \to \mathbb{R}$ over a compact domain $\mathcal{D} \subset \mathbb{R}^d$ that can be written as

$$\begin{aligned} f(x) &= \mathbb{E}_{\xi\sim\pi}[\ell(x,\xi)], \\ x^* &\in \arg\min_{x\in\mathbb{R}^d} f(x). \end{aligned} \tag{1}$$

where ξ denote samples from the data distribution π. In practice, gradient-based methods find the argmin by evaluating gradients $\nabla_x \ell(x,\xi)$. We assume $\ell(x,\xi)$ is differentiable in this work.

Nonconvex optimization comes with two typical problems for gradient-based methods. These methods often struggle to find the global minimum, getting stuck in saddle points or local minima. Extensive work has shown that stochastic noise can help SGD to escape saddle points [14, 20, 25], and in specific conditions, escape local minima [30, 38]. Separately, researchers may avoid gradient-based methods due to the difficulty or expense of gradient calculations. In some settings, the objective may not even be differentiable, for instance if given via a black-box procedure. This is common in applied fields such as structural engineering [23], climate modeling [32], geophysics [13], and other applied sciences where for instance Bayesian inversion plays a role. These fields have been at the forefront of developing gradient-free optimization methods. One of the most popular agent-based global optimization algorithms is Particle Swarm Optimization (PSO) [27]. In PSO, agents explore the state space while encountering a randomized drift pushing towards the global "best" position and their past personal "best" position. Developments in PSO led to the creation of Consensus Based Optimization (CBO) [7, 33, 39]. This algorithm switches off the personal best, and circumvents the selection of a "best" particle by treating all particles identically, which makes the dynamics amenable to analysis and has led to recent advances providing a rigorous theoretical framework for this method [18], whilst providing performance comparable to PSO [4, 21]. The method is a system of Stochastic Differential Equations (SDE) that mimics interacting agents communicating over a weighted mean. Particles are expected to build a consensus at the position of the weighted mean that is located near the global minimizer of the objective function.

In this study, we develop a hybrid framework that combines gradient-based and gradient-free methods. Our aim is to leverage the fast convergence benefits of gradient-based optimization along with the explorative benefits of the CBO algorithm. We especially focus on the cheaper compute of CBO, building an algorithm with high accuracy while adding a limited number of gradient evaluations. We devise this algorithm for a preset compute budget when limiting the number of gradient evaluations is required. We study the impact of changing this compute budget, studying the tradeoff between accuracy and compute cost. Our main contributions are as follows:

- We introduce a new hybrid particle-based algorithm for global non-convex optimization that takes advantage of both local gradient information and global consensus information to move an interacting particle system toward the minimizer;
- We show empirically that for ML Classification tasks, our hybrid algorithm attains similar accuracies as ensemble gradient methods (e.g., based on Gradient Descent and Adam) at a fraction of the computational cost.

2. Background

2.1. Consensus Based Optimization (CBO)

Consensus Based Optimization (CBO) was first introduced in [33]. It is a particle-based optimization method that computes a weighted mean using the ensemble. Particles with small function values have a larger influence on the weighted mean than those with large function values. Thus, the weighted mean is expected to approach the global minimizer of the objective function f.

All particles in the CBO algorithm are driven by two terms. One of these is a drift term, forcing the particles to move towards the weighted mean, which is also known as a consensus. This algorithm choice is motivated by the Laplace principle [33]. Given a measure $\rho \in \mathcal{M}(\mathbb{R}^d)$, we define the weighted mean of ρ as

$$v_f(\rho) := \int x\omega_\alpha^f(x)d\rho(x)\,, \qquad \omega_\alpha^f(x) := \frac{\exp(-\alpha f(x))}{\int \exp(-\alpha f(y))d\rho(y)}\,. \tag{2}$$

Note that here, the values of the objective function enter into the weight only. The mean is calculated such that particles with small function values have more influence in the weighted mean than those with large function values. The parameter α controls this separation effect and can be thought of as a control knob for explore vs exploit. For $\alpha = 0$, all particles have the same weight. As $\alpha \to \infty$, we expect v_f to approximate the global best of the agents, and can therefore be interpreted as a softmin of the objective over the particle positions. Particles in the CBO algorithm experience both a drift towards the weighted mean v_f, and a scaled diffusion which is dynamically switched off similar to simulated annealing.

The evolution of the particles $i = 1, \dots, J$ is modeled using the following SDE, where $\rho_J(t) := \sum_{i=1}^{J} \delta_{x^i(t)}$ represents the empirical distribution of particles at the given time t:

$$dx^i(t) = -\lambda(x^i(t) - v_f(\rho_J(t)))dt + \sqrt{2\sigma}|x^i(t) - v_f(\rho_J(t))|dW_t^i, \tag{3}$$

where λ scales the drift term and results in a faster attraction of particles to the mean, σ controls the intensity of the noise and W_t^i refers to the Brownian motion acting on particle i. Note that the drift and diffusion of the particle dynamics both scale with the distance from the weighted mean.

2.2. Related Work

For large enough λ and small enough σ, it is expected that eventually all particles collapse to the weighted mean v_f, which is often referred to as the *consensus point* of the ensemble. In [18], the authors provide probabilistic global convergence guarantees for the CBO algorithm. Recent work has proven that CBO can be considered a stochastic relaxation of Gradient Descent [35]. We use this reasoning as motivation for the selection of CBO to be deployed in combination with other gradient-based methods. Several adaptations of CBO have been proposed recently for high-dimensional machine learning problems [8], including random mini-batching [31], for constrained optimization problems [9, 11, 17, 22], multi-objective optimization [5, 29], for targets with multiple minimizers or distributions with many modes [6, 16], including momentum [12, 21, 24], memory [4, 40], truncated noise [19], and via jump-diffusion processes [26]. Recent applications include federated learning [10], finance [2], and rare event estimation [1].

Few variants have focused on including gradient information into traditional CBO. In [34] the authors, in addition to incorporating memory information, also evaluate local gradients, show global convergence in mean field law and demonstrate superior performance on benchmark optimization functions. In [37], an approximated gradient term is added to traditional CBO, observing convergence to "better minima" when tested on similar benchmark functions. Another line of work introduced the Adam-CBO method, which adds first and second order momentum terms to damp oscillation and accelerate convergence [12].

3. Algorithm Design

3.1. CBO Combined with Limited Gradient Evaluations

We develop an algorithm to combine the best of both CBO and gradient optimization, encouraging the inexpensive exploration of particles while also benefiting from the quick convergence of gradient optimization. We develop Algorithm 1 using N total particles and n particles to be updated using gradient information. Depending on the compute budget available, a suitable $n \leq N$ can be chosen. For each iteration, we select the n particles closest to the weighted mean. We assume that these particles are doing the 'best' in terms of approximating the global minimum, and so speeding up their performance accelerates the optimization for the full particle ensemble as this information is shared via the weighted mean. For these particles, we perform a gradient update step. The remaining $N - n$ particles are updated via the traditional CBO update step (3). The resulting algorithm allows for a combination of ensemble gradient methods and particle based methods. After T iterations, we return the best of all N particles.

3.2. Numerical Illustrations on a Toy Example

We study the performance of Algorithm 1 with Gradient Descent on a toy example with the objective $f(x) = (\frac{x}{2})^2 + 3(1 - \cos(2\pi x))$. This function has its global minima located at 0. We perform 100 independent runs and plot (i) their final best particle value on the function, and (ii) a histogram to understand their final distribution. All runs are initialized with 50 randomly spaced particles from [-40, 40], re-initializing particles for each run. Figure 1 illustrates that Gradient Descent (i.e., 100% gradient particles) gets stuck in local minima after T=1000 iterations, unlike CBO (i.e., 0% gradient particles). This demonstrates the benefits of the CBO algorithm, including its ability to explore a greater state space. With few iterations, we observe faster convergence of CBO with 50% gradient particles. In this case, more final particles are located at the global minimum as compared to CBO without gradients.

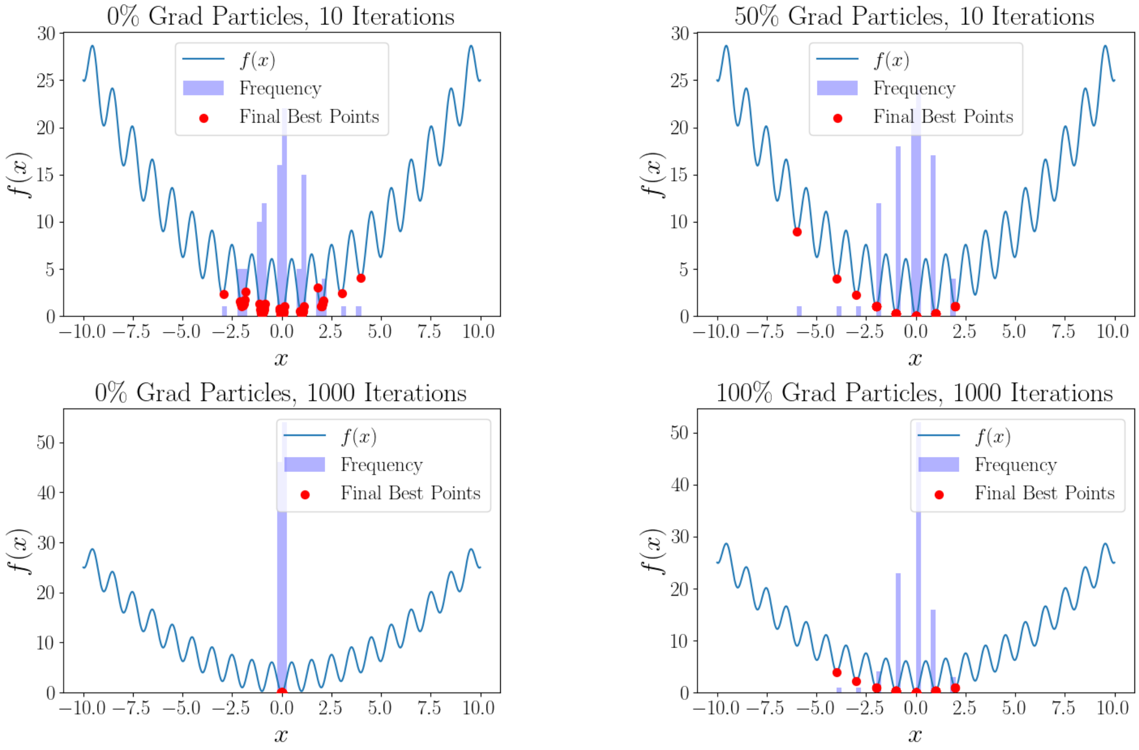

Figure 1: Distribution of final best particles (red) for 100 runs after T=10 iterations (top row) and T=1000 iterations (bottom row).

4. Numerical Results

For the following examples, we use `CBXpy` [3], replacing their CBO optimizer with Algorithm 1. At each iteration, we select particles closest to the weighted mean and update these using gradient information. For this high-dimensional problem, we test two algorithms, one using GD and one using Adam for updating the particles closest to the consensus. For details on initialization and hyperparameters see Appendix A.

```
Algorithm 1: CBO with Limited Local Gradient Evaluations
Input: Objective function $f$, number of particles $N$, number of particles for gradient
        evaluations $n$, number of iterations $T$, time step $\Delta t$, noise factor $\sigma$, learning rate $\lambda$,
        energy scaling factor $\alpha$, input domain $\mathcal{D}$
Output: Location of best particle, an approximation of $x^*$
Initialization: Randomly draw $N$ particles $x \in \mathbb{R}^d$ uniformly from the domain $\mathcal{D}$
for $t = 1$ to $T$ do
    Step 1: Compute Consensus
    a) Compute the energy of each particle: $E_i = f(x_i)$, $i = 1, \ldots, N$
    b) Calculate the weight of each particle: $w_i = \exp\left(-\alpha E_i - \log \sum_j \exp(-\alpha E_j)\right)$
    c) Compute consensus point: $c = \sum_i w_i x_i$
    Step 2: Gradient Descent and CBO Updates
    a) Compute distances from consensus point: $d_i = \|x_i - c\|$
    b) Identify the set of $n$ closest particles to the consensus $\mathcal{I}_{\text{GD}}$
    c) For all particles $i \in \mathcal{I}_{\text{GD}}$, apply the gradient update: $x_i \leftarrow x_i - \Delta t \nabla f(x_i)$
    d) For all particles $i \notin \mathcal{I}_{\text{GD}}$, apply the CBO correction with noise:
            $x_i \leftarrow x_i - \lambda \Delta t \cdot (x_i - \mathbf{c}) + \sqrt{2\sigma\Delta t}\|x_i - c\|\zeta, \qquad \zeta \sim \mathcal{N}(0, 1)$
end
Return the best particle: $x_{\text{best}} = \arg\min_i f(x_i)$
```

4.1. CBO with GD Particles

We use $N = 50$ total particles and vary the number of GD particles n from 0 to 50 in intervals of 5. Gradients are evaluated using Pytorch's automatic differentiation. We track the resulting accuracy associated with the best particle after 10 epochs. We additionally plot the time it takes to evaluate those 10 epochs. This is used to understand the tradeoff between number of gradient evaluations and accuracy. Results averaged over 10 runs are shown in Figure 2; we note that more runs yield similar results. As shown, the classification accuracy greatly increases with the addition of just a few GD particles. This benefit quickly plateaus. At the same time, adding more GD particles increases the computational cost linearly; see Figure 2 (right). From the slope of the linear increase in Figure 2 (right) one can extract the cost of one gradient evaluation for the MNIST dataset. In application settings where the cost of one gradient evaluation is known a priori, such analysis can be used to determine the optimal cost-accuracy trade-off. Our results suggest that if focused on minimizing computational cost, introducing just a few GD particles can yield improved accuracy.

4.2. CBO with Adam Particles

Due to memory limitations, we use $N = 30$ total particles and vary the number of Adam particles n from 0 to 30 at intervals of 5. We use the Pytorch implementation of the Adam optimizer. We similarly track the accuracy and time associated with running 7 epochs, averaged over 10 runs, of training the Neural Network. Figure 3 displays similar results to Figure 2, once again suggesting that if the focus is on minimizing computational cost, it may be useful to include minimal gradient information into the CBO algorithm, prioritizing the particles closest to the weighted mean.

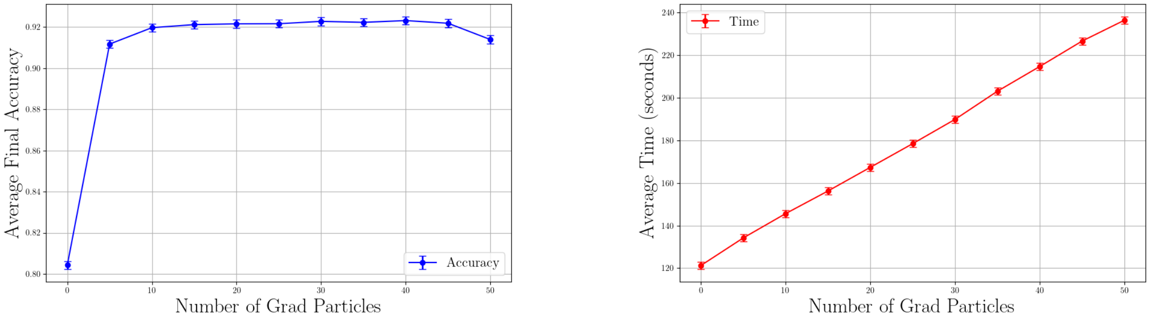

Figure 2: Accuracy and Time after 10 epochs, varying number of Gradient Descent particles.

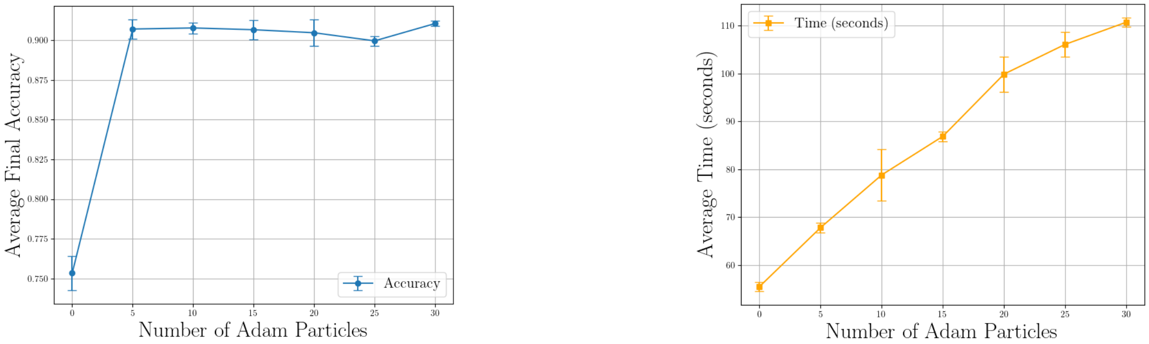

Figure 3: Accuracy and Time after 7 epochs, varying number of Adam particles.

5. Discussion

We propose a new algorithm based on Consensus Based Optimization that leverages gradient information. This is a novel approach for global optimization defined by an interacting particle system that uses both local gradient information and global information about the consensus of the particles. The inclusion of CBO methods can prevent particles from getting caught in local minima of a function, as the noise term in CBO encourages their exploration. We further demonstrate that for ML classification tasks, the proposed algorithm attains a similar accuracy to non-global ensemble gradient methods using both Gradient Descent and Adam, even for very few iterations, with significantly fewer gradient evaluations. We discuss the trade-off between accuracy and computation time. This algorithm can be a powerful tool in situations where evaluating gradients is costly. Given a limited computational budget, our algorithm selects a new set of particles at each iteration for gradient-based updates. This allows for accelerated accurate approximation of the desired minimum. Moreover, the only available theoretical guarantees for CBO apply in the asymptotic regime as time goes to infinity and do not account for algorithm behavior at early or intermediate times. In practice however, these methods are applied over a finite time horizon. Demonstrating effects of gradient information after short times is therefore crucial to develop accurate algorithms within a constrained computational budget. Future work will focus on more complex ML tasks as well as theoretical convergence guarantees with respect to the number of particles being updated via gradient evaluations.

6. Acknowledgments

AS was supported by the Kirk and Marjory Dawson Family SURF Fellowship and by start-up funds from California Institute of Technology. RB is grateful for support from the von Kármán instructorship at Caltech and a Department of Defense (DoD) Vannevar Bush Faculty Fellowship (award N00014-22-1-2790) held by Andrew M. Stuart.

References

[1] Konstantin Althaus, Iason Papaioannou, and Elisabeth Ullmann. Consensus-based rare event estimation. *SIAM Journal on Scientific Computing*, 46(3):A1487–A1513, 2024.

[2] Hyeong-Ohk Bae, Seung-Yeal Ha, Myeongju Kang, Hyuncheul Lim, Chanho Min, and Jane Yoo. A constrained consensus based optimization algorithm and its application to finance. *Applied Mathematics and Computation*, 416:126726, 2022.

[3] Rafael Bailo, Alethea Barbaro, Susana N Gomes, Konstantin Riedl, Tim Roith, Claudia Totzeck, and Urbain Vaes. Cbx: Python and julia packages for consensus-based interacting particle methods. *arXiv preprint arXiv:2403.14470*, 2024.

[4] Giacomo Borghi, Sara Grassi, and Lorenzo Pareschi. Consensus based optimization with memory effects: random selection and applications. *Chaos, Solitons & Fractals*, 174:113859, 2023.

[5] Giacomo Borghi, Michael Herty, and Lorenzo Pareschi. An adaptive consensus based method for multi-objective optimization with uniform pareto front approximation. *Applied Mathematics & Optimization*, 88(2):58, 2023.

[6] Leon Bungert, Tim Roith, and Philipp Wacker. Polarized consensus-based dynamics for optimization and sampling. *Mathematical Programming*, pages 1–31, 2024.

[7] José A Carrillo, Young-Pil Choi, Claudia Totzeck, and Oliver Tse. An analytical framework for consensus-based global optimization method. *Mathematical Models and Methods in Applied Sciences*, 28(06):1037–1066, 2018.

[8] José A Carrillo, Shi Jin, Lei Li, and Yuhua Zhu. A consensus-based global optimization method for high dimensional machine learning problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 27:S5, 2021.

[9] José A Carrillo, Shi Jin, Haoyu Zhang, and Yuhua Zhu. An interacting particle consensus method for constrained global optimization. *arXiv preprint arXiv:2405.00891*, 2024.

[10] José A Carrillo, Nicolas Garcia Trillos, Sixu Li, and Yuhua Zhu. Fedcbo: Reaching group consensus in clustered federated learning through consensus-based optimization. *Journal of machine learning research*, 25(214):1–51, 2024.

[11] José Antonio Carrillo, Claudia Totzeck, and Urbain Vaes. Consensus-based optimization and ensemble kalman inversion for global optimization problems with constraints. In *Modeling and Simulation for Collective Dynamics*, pages 195–230. World Scientific, 2023.

[12] Jingrun Chen, Shi Jin, and Liyao Lyu. A consensus-based global optimization method with adaptive momentum estimation. *arXiv preprint arXiv:2012.04827*, 2020.

[13] Mohsen Dadashpour, David Echeverria Ciaurri, Tapan Mukerji, Jon Kleppe, and Martin Landrø. A derivative-free approach for the estimation of porosity and permeability using time-lapse seismic and production data. *Journal of Geophysics and Engineering*, 7(4):351–368, 2010.

[14] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In *International Conference on Machine Learning*, pages 1155–1164. PMLR, 2018.

[15] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[16] Massimo Fornasier and Lukang Sun. A pde framework of consensus-based optimization for objectives with multiple global minimizers. *arXiv preprint arXiv:2403.06662*, 2024.

[17] Massimo Fornasier, Hui Huang, Lorenzo Pareschi, and Philippe Sünnen. Consensus-based optimization on hypersurfaces: Well-posedness and mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 30(14):2725–2751, 2020.

[18] Massimo Fornasier, Timo Klock, and Konstantin Riedl. Consensus-based optimization methods converge globally. *SIAM Journal on Optimization*, 34(3):2973–3004, 2024.

[19] Massimo Fornasier, Peter Richtárik, Konstantin Riedl, and Lukang Sun. Consensus-based optimisation with truncated noise. *European Journal of Applied Mathematics*, pages 1–24, 2024.

[20] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.

[21] Sara Grassi and Lorenzo Pareschi. From particle swarm optimization to consensus based optimization: stochastic modeling and mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 31(08):1625–1657, 2021.

[22] Seung-Yeal Ha, Myeongju Kang, Dohyun Kim, Jeongho Kim, and Insoon Yang. Stochastic consensus dynamics for nonconvex optimization on the stiefel manifold: Mean-field limit and convergence. *Mathematical Models and Methods in Applied Sciences*, 32(03):533–617, 2022.

[23] Warren Hare, Julie Nutini, and Solomon Tesfamariam. A survey of non-gradient optimization methods in structural engineering. *Advances in Engineering Software*, 59:19–28, 2013.

[24] Hui Huang, Jinniao Qiu, and Konstantin Riedl. On the global convergence of particle swarm optimization methods. *Applied Mathematics & Optimization*, 88(2):30, 2023.

[25] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pages 1724–1732. PMLR, 2017.

[26] Dante Kalise, Akash Sharma, and Michael V Tretyakov. Consensus-based optimization via jump-diffusion stochastic differential equations. *Mathematical Models and Methods in Applied Sciences*, 33(02):289–339, 2023.

[27] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995. doi: 10.1109/ICNN.1995.488968.

[28] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] Kathrin Klamroth, Michael Stiglmayr, and Claudia Totzeck. Consensus-based optimization for multi-objective problems: a multi-swarm approach. *Journal of Global Optimization*, pages 1–32, 2024.

[30] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? In *International conference on machine learning*, pages 2698–2707. PMLR, 2018.

[31] Dongnam Ko, Seung-Yeal Ha, Shi Jin, and Doheon Kim. Convergence analysis of the discrete consensus-based optimization algorithm with random batch interactions and heterogeneous noises. *Mathematical Models and Methods in Applied Sciences*, 32(06):1071–1107, 2022.

[32] Ignacio Lopez-Gomez, Costa Christopoulos, Haakon Ludvig Langeland Ervik, Oliver RA Dunbar, Yair Cohen, and Tapio Schneider. Training physics-based machine-learning parameterizations with gradient-free ensemble kalman methods. *Journal of Advances in Modeling Earth Systems*, 14(8):e2022MS003105, 2022.

[33] René Pinnau, Claudia Totzeck, Oliver Tse, and Stephan Martin. A consensus-based model for global optimization and its mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 27(01):183–204, 2017.

[34] Konstantin Riedl. Leveraging memory effects and gradient information in consensus-based optimisation: On global convergence in mean-field law. *European Journal of Applied Mathematics*, pages 1–32, 2023.

[35] Konstantin Riedl, Timo Klock, Carina Geldhauser, and Massimo Fornasier. Gradient is all you need? *arXiv preprint arXiv:2306.09778*, 2023.

[36] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[37] Claudia Schillings, Claudia Totzeck, and Philipp Wacker. Ensemble-based gradient inference for particle methods in optimization and sampling. *SIAM/ASA Journal on Uncertainty Quantification*, 11(3):757–787, 2023.

[38] Sebastian Stich and Harsh Harshvardhan. Escaping local minima with stochastic noise. In *Neurips Workshop on Optimization for Machine Learning*, 2021.

[39] Claudia Totzeck. Trends in consensus-based optimization. In *Active Particles, Volume 3: Advances in Theory, Models, and Applications*, pages 201–226. Springer, 2021.

[40] Claudia Totzeck and Marie-Therese Wolfram. Consensus-based global optimization with personal best. *arXiv preprint arXiv:2005.07084*, 2020.

Appendix A. Details on numerical experiments

For experiments in Section 3.2, we set the parameters as $dt = 0.01$, $\sigma = 1.0$, $\lambda = 1.0$, $\alpha = 5.0$. For this one dimensional problem, we set $M = 1, N = 50, d = 1$.

For the Neural Network experiments in Section 4, we use the setup described in the `CBXpy` documentation [33]. This includes $dt = 0.1$, $\sigma = 0.1$, $\lambda = 1.0$, $\alpha = 50.0$ and uses $N = 50$ total particles. Timing experiments in Section 4 were run on an Apple Silicon M1 chip with a 3.2 GHz processor.

We replicate the task of classifying the MNIST dataset of handwritten digit images [15] by measuring the time required for computation and the classification accuracy after 10 epochs. The neural network architecture we consider consists of two layers: an input layer with 784 units and an output layer with 10 units. Each layer is followed by a ReLU activation function and batch normalization. A softmax activation function is applied after the final layer to estimate the probability of each label.

A.1. Discussion of Particle Error

We use the function $f(x) = (\frac{x}{2})^2 + 3(1 - \cos(2\pi x))$ from the numerical illustrations in Section 3.2. In the following illustration, we plot the error of the best particle after 1000 iterations, which demonstrates an increase with the number of gradient particles. The results are averaged over 100 runs.

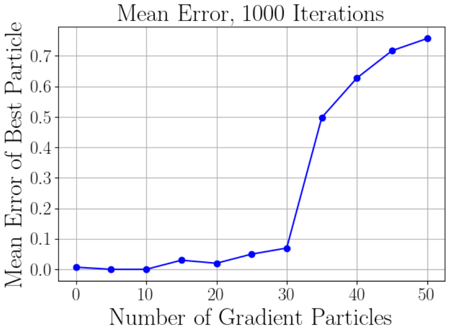

Figure 4: Error of best particle after 1000 iterations

A.2. Comparison with Ensemble Methods

We compare the time and accuracy for three different splits of the particles updated using CBO or Adam. Table 1 presents the results after 7 epochs by averaging over 10 iterations. The setup for the neural network architecture is the same as described in Section 4. N refers to the total number of particles, and n refers to the number of particles evaluated using the Adam update (rather than CBO) at each iteration.

	Average Accuracy	Average Time (s)
N=10, n=10	0.88764	57.0201
N=30, n=10	0.90732	84.8080
N=30, n=30	0.91057	110.7147

Table 1: Average accuracy and time from running Algorithm 1 for different splits of N particle into n Adam and $N - n$ CBO updates. Note that experiments with $N = 30, n = 10$ and $N = 30, n = 30$ have similar accuracies. However, $N = 30, n = 10$ requires less time because it involves fewer gradient evaluations.